REMARKS/ARGUMENTS

Claims 1-4, 7, 8, 10-14 and 17-24 are pending in the present application. Claims 1, 10, 11 and 18 have been amended, and Claims 21-24 have been added, herewith. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 101

Claims 11-14 and 17-20 stand rejected under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. This rejection is respectfully traversed.

With respect to Claim 11, Applicants have amended such claim in accordance with the Specification description at page 11, lines 5-8, to explicitly recite hardware elements.

Applicants traverse the rejection of Claims 12-14 for reasons given above with respect to Claim 11 (of which Claims 12-14 depend upon).

Claim 18 fully complies with the USPTO's guidelines regarding proper statutory subject matter. For example:

> "When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994)(claim to data structure stored on a computer readable medium that increases computer efficiency held statutory) and *Warmerdam*, 33 F.3d at 1360-61, 31 USPQ2d at 1760 (claim to data structure *per se* held nonstatutory)" (emphasis added by Applicants).

Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility. [1]

Claim 18 recites "A computer readable medium encoded with a computer program product that is operable in a data processing system for monitoring transactions for a set of known nodes in a network data processing system". Applicants urge that a computer readable medium encoded with a computer program product that is operable in a data processing system for monitoring transactions for a set of known nodes in a network data processing system is a computer element which defines structural and functional inter-relationships between the computer program and the rest of the computer which permits the computer program's functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84,

[1] http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelines101_20051026.pdf

32 USPQ2d at 1035.[2] Accordingly, as Claim 18 expressly recites a computer readable medium encoded with a computer program product that is operable in a data processing system for monitoring transactions for a set of known nodes in a network data processing system, it is shown that Claim 18 (and similarly for Claims 19 and 20) is directed to statutory subject matter, pursuant to both judicial case law and the USPTO's own MPEP rules. Thus, Claim 18 is statutory under 35 U.S.C. § 101.

Still further, Claim 18 explicitly recites a computer readable medium encoded with a computer program product that is operable in a data processing system for monitoring transactions for a set of known nodes in a network data processing system, which is either a 'manufacture' or a 'composition of matter', both of which are statutorily recognized subject matter[3]. In addition, since Claim 18 explicitly recites a computer readable medium encoded with a computer program product that is operable in a data processing system for monitoring transactions for a set of known nodes in a network data processing system, such claim does *not* fall within one of the three judicially determined exceptions of: natural phenomenon, law of nature or abstract idea (see, e.g., MPEP 2106 and in particular MPEP 2106(IV)(B) and (C)), but instead is limited to a practical application in the technological arts[4]. Thus, it is further shown that Claim 18 is allowable in view of 35 U.S.C. § 101 as the invention recited therein does not fall within a judicial exception but instead is limited to a practical application in the technological arts.

It is further urged that Claim 18 is very different from the type of claim rejected under 35 U.S.C. § 101 in *In re Nuijten*, 84 USPQ2d 1495, in that such Nuijten claim was specifically directed to operations (watermarking) performed on a data signal itself ("A method of embedding supplemental data in a signal…encoding the signal…modifying selected samples of the encoded signal"). As described above, Claim 18 is not directed to operations being performed on a signal itself, and thus the holding in *In re Nuijten, Id.* is not applicable to Claim 18. For example, this same Nuijten patent application had a

[2] The USPTO's own guidelines similarly state this type of claim is proper under 35 U.S.C. § 101. For example, as stated in the Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility (published in the Official Gazette on November 22, 2005) at ANNEX IV (Computer-Related Nonstatutory Subject Matter) "When functional descriptive material is recorded on some **computer-readable medium** it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994) (claim to data structure stored on a computer readable medium that increases computer efficiency held statutory) and *Warmerdam*, 33 F.3d at 1360-61, 31 USPQ2d at 1759 (claim to computer having a specific data structure stored in memory held statutory product-by-process claim) with *Warmerdam*, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure per se held nonstatutory)."

[3] **35 U.S.C. 101 Inventions patentable.**

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

[4] *Only when* the claim is devoid of any limitation to a practical application in the technological arts should it be rejected under 35 U.S.C. § 101. Compare *Musgrave*, 431 F.2d at 893, 167 USPQ at 289; *In re Foster*, 438 F.2d 1011, 1013, 169 USPQ 99, 101 (CCPA 1971).

program product claim (Claim 15) that was not the subject of appeal, *and this program product claim was allowed*, *In re Nuijten, Id.*

Therefore, the rejection of Claims 11-14 and 17-20 under 35 U.S.C. § 101 has been overcome.

II. 35 U.S.C. § 102, Anticipation

Claims 1-4, 7, 8, 10-14 and 17-20 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Feldmann et al. (U.S. Publication No. 2002/0103631 A1), hereinafter "Feldmann". This rejection is respectfully traversed.

Generally, the present invention is directed to a method, system and program product that provides load balancing based on actual transaction data that is identified using data obtained from router caches. In contrast, the cited reference is directed to a traffic engineering system where a data model is used to model a network – and in fact this cited reference expressly disparages use of actual network load-based data, such as underlying transactions, in their analysis due to difficulties in processing such type of data, as will now be described in detail.

With respect to Claim 1, such claim recites "identifying the transactions handled by each node in the set of known nodes using the identification of the set of nodes included in the cache data received from the router, to form identified transactions". As can be seen, per the features of Claim 1, transactions that are handled by each node are identified, and such transaction identification is performed using the cache data received from the router. In rejecting this aspect of Claim 1, the Examiner cites Feldmann's descriptions at Abstract, paragraphs 20, 36, 52, 53, claim 15, claim 32 and Figures 3-5 as teaching such transaction identification for each node using cache data. Applicants urge error, as follows.

The cited passage at Feldmann's Abstract describes a novel system and method for traffic engineering in a packet switching network, where a global view of a network is constructed using a network model, where such network model is constructed from 'network information' associated locally with the individualized elements in the network. This 'network information' is not further described in the Abstract, and thus the Abstract does not teach the claimed step of "*identifying the transactions handled by each node* in the set of known nodes *using the cache data received from the router*, to form identified transactions".

The cited passage at Feldmann's paragraph [0020] describes a network data 'model', including representations for routers and access links connected to the routers, and backbone links that connect the routers inside the ISP backbone. However – and importantly – such model does not contain information regarding specific transactions for nodes, instead maintaining information regarding identification information pertaining to the network *links* themselves (Feldmann paragraph [0021], "networks"). Restated, this Feldman description describes network *topology* information (by analogy, such topology information is similar to a highway 'map' (also see Feldmann paragraph [0018])), and not network *transaction*/load information (by analogy, similar to highway 'traffic' (also see Feldmann paragraph [0034])).

The cited passage at Feldmann's paragraph [0036] describes use of a data 'model' that is used to model network traffic, where *traffic demands between routers* are represented as data objects. No use of actual router cache data is used to identify specific transactions for each of the nodes. Importantly, Feldmann *expressly teaches away from using transaction data* in its traffic monitoring, due to: (i) the enormous amount of data that would be required to be processed, (ii) there is no common source/point for obtaining transaction data, and (iii) difficulties in interpreting inter-domain traffic (Feldmann paragraph [0035]). This 'teaching away' passage at Feldermann paragraph [0035] – due to its critical importance in interpreting the teachings of the cited Feldmann reference - is reproduced herein for ease of access:

> "[0034] B. Traffic Demand
>
> [0035] Effective traffic engineering requires not just a view of the topology but also an accurate estimate of the offered load between various points in the backbone. How should traffic demands be *modeled* and *inferred* from operational measurements? At one extreme, **IP traffic *could* be represented at the level of individual source-destination pairs, possibly aggregating sources and destinations to the network address or AS level. Representing all hosts or network addresses, however, would result in an overly large traffic matrix, virtually impossible to populate since no single ISP is likely to see all of the traffic to and from each network address.** Alternatively, IP traffic demands might be aggregated to point-to-point demands between edge links or routers in the ISP backbone. This approach, however, has fundamental difficulties in dealing with interdomain traffic (traffic whose ultimate destination belongs to another domain). Inter-domain traffic, which constitutes a large fraction of traffic in operational IP networks today, may exit the ISP backbone from any of a set of egress links, determined by interdomain routing policies. **Modeling interdomain traffic as point-to-point would couple the demand model to internal routing configuration, making it highly problematic to predict how changing internal routing configuration would influence network load**" (emphasis added by Applicants).

So instead of using actual transaction data for determining traffic/load, Feldmann instead uses an 'alternative model' of traffic demand that consists of an ingress link, a set of egress links, and a volume of load, where the volume of load is an amount of data for a given demand, represented as a raw number (in kilobytes), as described by Feldmann at paragraph [0036]. Instead of collecting packet-level traces (as per the features of Claim 1, since the router cache data is for specific, individual transactions as described in the Specification at page 14, lines 14-20), the cited reference expressly teaches away from such technique (Feldmann paragraph [0039], lines 4-6) by using an alternative Netflow technique that keeps track of the amount of traffic in each active flow (Feldmann paragraph [0039], lines 5-17). This 'amount of traffic' for a flow between routers does not provide any type of specific, individual transaction identification capability, as per the features of Claim 1.

The cited passage at Feldmann's paragraph [0052] describes a routing model, whereby (i) traffic between two routers in the same area follows a shortest path within the area, and (ii) traffic between routers in different areas follows a shortest path without regard to area boundaries. Importantly, this 'shortest path' is performed using a shortest-path-first algorithm, and is not based-on and does not use actual transactions for each of the nodes, where such transactions are identified using cache data received from a router, as per the features of Claim 1.

The cited passage at Feldmann's paragraph [0053] describes a technique for resolving a tie situation when the above described shortest path algorithm results in multiple shortest path solutions. In such a tie situation, load-balancing is achieved by performing a hash function on a source and destination IP address *of each packet*. This hashing of information contained in a data packet does not teach *identification of transactions for <u>each node</u> using <u>cache data</u> received from a cache.*

The cited passage at Feldmann's claims 15 and 32 both describe the graphical display of network traffic data *<u>calculated</u> by the routing <u>model</u>.* This traffic modeling technique does not teach identification of transactions for each node using cache data received from a cache.

Finally, the cited Feldmann Figures 3-5 do not depict any type of transaction identification, either as per the specific features of Claim 1, or otherwise. Instead, Figures 3-5 depict algorithms used by the modeling technique to analyze traffic flow on links based on 'flow records' (Feldmann paragraph [0040]). For example, Figure 3 calculates an overall 'volume of traffic' for a given flow for a given time period (see lines 1 and 8 of Figure 3, for example). This overall volume of traffic for a given flow calculation does not teach identification of transactions for each node using cache data received from a cache.

It should be further noted that Feldmann's description of a router configuration file and forwarding table are not equivalent to the claimed router cache or associated data, as the Feldmann configuration file merely contains topology configuration information, and not specific transaction-identifying information (Feldmann paragraph [0029]), and the Feldmann router table is used to determine a 'next-hop' IP address, and therefore does not include specific transaction-identification information (Feldmann paragraph [0031]; see also ATTACHMENT I and ATTACHMENT II attached hereto to further describe routing tables and their use in routers – note in particular pages 15, 18 and 19).

Therefore, Claim 1 has been erroneously rejected as every element recited in such claim is not identically shown in a single reference.

Applicants initially traverse the rejection of Claims 2-4, 7 and 8 for reasons given above with respect to Claim 1 (of which Claims 2-4, 7 and 8 depend upon).

Further with respect to Claim 2, such claim recites "wherein the cache data is from an address resolution protocol cache located on the router". As can be seen, the cache data that is used for identifying transactions is from an address resolution protocol cache located on the router. The cited reference does not teach or otherwise describe any type of an address resolution protocol cache that is located on a router, and thus it is further urged that Claim 2 has been erroneously rejected due to this additional missing claimed feature that is not identically shown in the cited reference.[5]

Applicants initially traverse the rejection of Claims 10-14 and 17-20 for similar reasons to those given above with respect to Claim 1.

Applicants further traverse the rejection of Claims 12 and 19 for similar reasons to the further reasons given above with respect to Claim 2.

Therefore, the rejection of Claims 1-4, 7, 8, 10-14 and 17-20 under 35 U.S.C. § 102(b) has been overcome.

III. Newly Added Claims

Claims 21-24 have been added herewith, in accordance with the Specification description at page 9, line 15 – page 10, line 16, page 13, lines 21-28, and page 17, lines 3-8. Examination of such claims is respectfully requested.

IV. Conclusion

It is respectfully urged that the subject application is patentable over Feldmann and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: August 13, 2008

Respectfully submitted,

/Wayne P. Bailey/
Wayne P. Bailey
Reg. No. 34,289
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

[5] For a prior art reference to anticipate in terms of 35 U.S.C. 102, *every element* of the claimed invention *must be identically shown* in a single reference. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990) (emphasis added by Applicants).

ATTACHMENT I

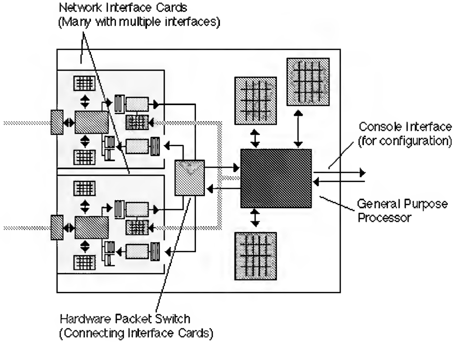
Operation of a Router

(source: http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/router-opn.html)

A modern router is a complex piece of equipment. The outside of the equipment is usually very simple, consisting of a number of network interface ports (shown on the left in orange in the figure below) to which cables may be connected and a few indicator lights to indicate that the router is functional.

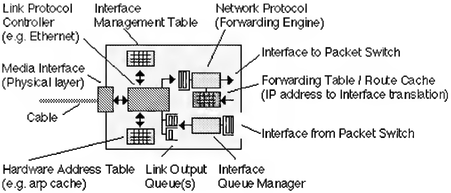
Overview

Most routers also have a serial connector to which a terminal (or a modem) may be connected, known as the "Console Port" (shown to the right in the figure below). This port is usually used to control the router configuration when the router is first installed. It may be the only port which is allowed to configure the filter table (used to prevent unauthorised access between the connected networks).



Block diagram of a complete router showing the interfaces Interfaces to the network (over which packets are received or transmitted) are shown left, and the control interface (used to set up and reconfigure the router) is shown right.

In the simplest case, the processing of packets is implemented in the general purpose processor which implements all the algorithms. More advanced routers may separate "forwarding" (the tasks of moving packets from one interface to another) from "routing" (the task of determining the best path through the network) and include a number of processors capable of performing these tasks. A router interface card resembles the LAN Network Interface Cards (NICs) used in PCs except that the card is normally of a higher specification (faster packet processing). The very first routers were designed used standard network interface cards, but modern high performance routers use special high performance interface cards and may also include a "Forwarding Engine" on-board the card which speeds the operation.


Link Protocol Controller (e.g. Ethernet)
Interface Management Table
Network Protocol (Forwarding Engine)
Interface to Packet Switch
Media Interface (Physical layer)
Forwarding Table / Route Cache (IP address to Interface translation)
Cable
Interface from Packet Switch
Hardware Address Table (e.g. arp cache)
Link Output Queue(s)
Interface Queue Manager

Router Network Interface Card

Received packets are processed by the link layer protocol controller, which handles the link layer protocol (e.g. HDLC, Ethernet) used over the physical link (cable). This also checks the received frame integrity (size, checksum, address, etc). Valid frames are converted to packets by removing the link layer header and are queued in the receive queue. This is usually a First-In-First-Out (FIFO) queue, often in the form of a ring of memory buffers.

The buffers are passed (drained) into the input to the forwarding engine. This takes each buffer, one at a time, and removes it from the interface receiver. The packet is then forwarded to an appropriate output interface, corresponding to the "best" path to the destination specified in the destination address of the IP packet header.

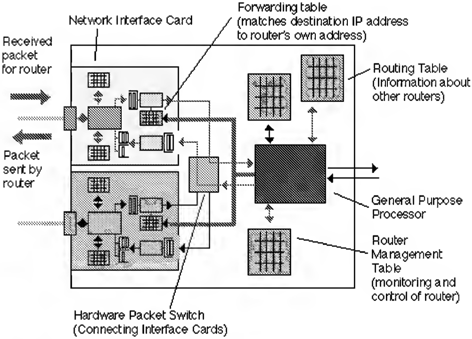
At the output interface, the packet (together with a new link layer header) is placed into a transmit queue until the link layer processor is ready to transmit the packet. This, like the receive queue, is a FIFO queue, and usually also takes the form of a ring of memory buffers.

Each out-going packet requires a new link layer protocol header to be added (encapsulation) with the destination address set to the next system to the receive the packet. The link protocol controller also maintains the hardware address table associated with the interface. This usually involves using the Address Resolution Protocol (arp) to find out the hardware (Medium Access Control) addresses of other computers or routers directly connected to the same cable (or LAN). The packet is finally sent using the media interface with the hardware address set to the next hop system. When complete, the buffer (memory) allocated to the frame, is "freed", that is, it is returned as an empty buffer to the receive queue, where it may be used to store a new received packet.

You may think from this that the job of forwarding is not too difficult, and involves a lot of copying of the packet data from one place to another. You would be wrong on both counts! Forwarding actually involves lots of decisions. Modern routers avoid copying the data in a packet if at all possible - this is a significant processing cost, and may easily slow down a router to a very low throughput. Instead, where ever possible, the router will leave the packet data in the same place and instead pass information about *where* a packet is stored in memory.

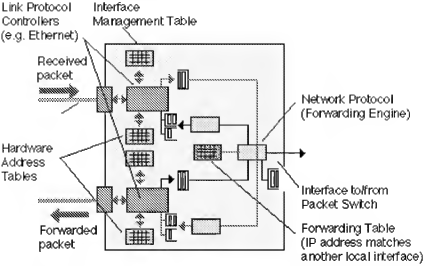
Introduction to Forwarding

This section gives a simple description of the forwarding process. After determining the link layer frame is valid, the forwarding engine then starts processing the network layer information. It reads the network layer (IP) packet headers and checks various parts of the header, to ensure the packet is not damaged or illegal. It then uses a local **Forwarding Table** (known as the "**Forwarding Information Base (FIB)**") to identify where in the network the packet should be routed to (i.e. which output interface should be used).



Forwarding a received packet to an output interface via the packet switch.

Once the appropriate output interface has been identified, the forwarding engine then requests the packet switch to form a connection to the appropriate output interface. The packet is then moved through the router to the output network interface controller. Although large routers actually implement a switch as a hardware component, most smaller routers do not actually contain a "real" switch. In other routers, the switch takes the form of a shared memory data structure in which all received packets are stored. The switching operation therefore consists of removing a pointer from the receive queue, and copying the value of the pointer to the appropriate transmit queue. In some cases, the entire packet data is copied from one bank of receive memory to another transmit memory using a computer bus.


Link Protocol Controllers (e.g. Ethernet)
Interface Management Table
Received packet
Network Protocol (Forwarding Engine)
Hardware Address Tables
Interface to/from Packet Switch
Forwarded packet
Forwarding Table (IP address matches another local interface)

Packets may be directly forwarded from one controller to the other (sometimes known as "Fast Switching"). This occurs when the forwarding table has a match for the IP destination address of the received packet which indicates the packet should be sent out using one of the other interfaces on the same card, and the packet does not require any special IP processing. This type of forwarding is very efficient, since it causes very little load on the router processor.

ATTACHMENT II

Routing Tables

(source: http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/intwork/inac_uni_lbxy.mspx?mfr=true)

During the routing process, the routing decisions of hosts and routers are aided by a database of routes known as the routing table. The routing table is not exclusive to a router. Depending on the routable protocol, hosts may also have a routing table that may be used to decide the best router for the packet to be forwarded. IP hosts have a routing table. IPX hosts do not have a routing table.

The types of possible entries in a routing table include:

Network Route . A route to a specific Network ID in the internetwork.

Host Route . A route to a specific internetwork address (Network ID and Host ID). Instead of making a routing decision based on just the network ID, the routing decision is based on the combination of network ID and host ID. Host routes allow intelligent routing decisions to be made for each internetwork address. Host routes are typically used to create custom routes to control or optimize specific types of internetwork traffic.

Default Route . A route that is used when no other routes for the destination are found in the routing table. For example, if a router or end system cannot find a network route or host route for the destination, the default route is used. Rather than being configured with routes for all the Network IDs in the internetwork, the default route is used to simplify the configuration of end systems or routers.

Note

In many router implementations including the Windows 2000 Routing and Remote Access service, there is a routing table and a forwarding table. The routing table is used to store all the routes from all possible sources. The forwarding table is what is used by the routable protocol when forwarding the packet. For example, for a Windows 2000 router, the Routing and Remote Access service maintains the IP routing table using a component called the Route Table Manager. The IP forwarding table is contained within the TCP/IP protocol. The Route Table Manager updates the IP forwarding table based on incoming route information from multiple sources. The contents of the routing table do not necessarily match the contents of the forwarding table. For the purposes of discussion in this introductory chapter, the routing table and the forwarding table are the same.

Routing Table Structure

As illustrated in Figure 1.5, entries in the routing table usually consist of the following fields:

Network ID The Network ID field contains the identification number for a network route or an internetwork address for a host route.

Forwarding Address The Forwarding Address field contains the address to which the packet is to be forwarded. The forwarding address can be a network interface card address or an internetwork address. For network IDs to which the end system or router is directly attached, the Forwarding Address field may be blank.

Interface The Interface field indicates the network interface that is used when forwarding packets to the network ID. This is a port number or other type of logical identifier. For example, the interface for a 3COM EtherLink III network interface card may be referred to as ELNK3 in the routing table.

Metric The Metric field indicates the cost of a route. If multiple routes exist to a given destination network ID, the metric is used to decide which route is to be taken. The route with the lowest metric is the preferred route. Some routing algorithms only store a single route to any Network ID in the routing table even when multiple routes exist. In this case, the metric is used by the router to decide which route to store in the routing table.

Metrics can indicate different ways of expressing a route preference:

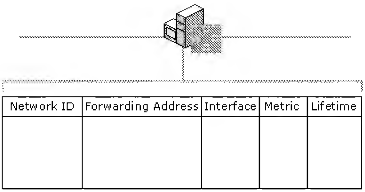
Hop Count . A common metric. Indicates the number of routers (hops) in the path to the network ID.

Delay . A measure of time that is required for the packet to reach the network ID. Delay is used to indicate the speed of the path—local area networks (LAN) links have a low delay, wide area network (WAN) links have a high delay—or a congested condition of a path.

Throughput . The effective amount of data that can be sent along the path per second. Throughput is not necessarily a reflection of the bit rate of the link, as a very busy Ethernet link may have a lower throughput than an unutilized 64-Kbps WAN link.

Reliability . A measure of the path constancy. Some types of links are more prone to link failures than others. For example, with WAN links, leased lines are more reliable than dial-up lines.

Lifetime The Lifetime field indicates the lifetime that the route is considered valid. When routes are learned through the exchange of information with other routers, this is an additional field that is used. Learned routes have a finite lifetime. To keep a learned route in the routing table, the route must be refreshed through a periodic process. If a learned route's lifetime expires, it is removed from the routing table. The timing out of learned routes provides a way for routers to reconfigure themselves when the topology of an internetwork changes due to a downed link or a downed router.



Network ID	Forwarding Address	Interface	Metric	Lifetime

Figure 1.5 Routing Table Structure

Note

The Lifetime field is typically not visible in routing tables.

This list of fields is a representative list in the routing tables. Actual fields in the routing tables for different routable protocols may vary. For information about the IP routing table, see "Introduction to TCP/IP" in the *TCP/IP Core Networking Guide* . For information about the IPX routing table, see "IPX Routing" in this book.

Top of page

Locality of the Routing Table

All the routing decisions made by the end system or the router are based on information in a local routing table that physically resides in the random access memory (RAM) of the system making the routing decision. There is no single, holistic view of the internetwork that is being gathered by a server and downloaded to each end system and router so that all users have the same view of the internetwork and all traffic flows along predictable pathways.

Each router in a path between a source and destination makes a local routing decision based on its local routing table. The path taken from the source to the destination may not be the same as the path for response packets from the destination back to the source. If the information in the local routing tables of the end systems or routers is incorrect due to misconfiguration or changing network conditions, then routing problems can result. Troubleshooting routing problems may involve the analysis of the routing tables of the end systems (source and destination) and all the routers forwarding packets between them.